**Action Plan**

**Phase 1: Update Core Logic (main.js)**

You will modify the processTraitString function in main.js to implement the new, refined token rules.

1. **Remove Obsolete Tokens**: Eliminate all logic related to {a\_name} and {the\_name}.
2. **Implement New {name} Logic**:
   * The function should check the NPC's isUnique and isProperName flags.
   * If isProperName is true, the token will be replaced by only the **first word** of the NPC's name, with the first letter capitalized (e.g., "Maasq Hammerheart" becomes "Maasq").
   * If isUnique is true (but isProperName is false), the token will be replaced by the NPC's full name, with the first letter capitalized.
   * If neither flag is true, the token will be replaced by "the " followed by the NPC's full name in lowercase (e.g., "the goblin").
3. **Handle Capitalization**:
   * The function must handle both lowercase ({name}, {he}) and capitalized ({Name}, {He}) versions of tokens.
   * When a capitalized token like {Name} is found, it should apply the same logic as {name} but ensure the final output is capitalized (e.g., "The goblin", "Maasq").
   * The pronoun tokens ({he}, {him}, {his}, {himself}) and their capitalized counterparts must continue to function correctly based on the NPC's gender.

**Phase 2: Fix Viewport Bug (viewport.js)**

You will update viewport.js to ensure that token replacement works in the main description field, not just in the traits.

1. **Locate Description Rendering**: In the updateViewport function, find the line where the description variable is used to build the final HTML.
2. **Process Description Content**: Before rendering the description, pass its content through the window.app.processTraitString function. This will ensure any tokens within the main description text are correctly replaced according to the logic defined in Phase 1.

**Phase 3: Update Modal UI (npc engineer.html)**

You will modify the "Manage Traits" modal in npc engineer.html to reflect the new interaction model.

1. **Update Quick Reference**: Find the trait-reference-content div.
2. **Simplify Content**: Remove the existing token explanations. Replace them with a clear, concise guide explaining that clicking a button inserts a lowercase token, while **Shift+Clicking** the button inserts the capitalized version.
3. **Remove Toggle**: Delete the <a> tag with the ID trait-reference-toggle. The reference will now be permanently visible, so the "Show/Hide" link is no longer necessary.

**Phase 4: Implement UI Interactivity (ui.js)**

You will update ui.js to add the new shift-click functionality to the token buttons in the "Manage Traits" modal.

1. **Modify Event Listener**: In the setupTraitListeners function, locate the event listener attached to modalTokenButtons.
2. **Detect Shift Key**: Modify the listener's callback function to check the shiftKey property of the click event object.
3. **Implement Conditional Logic**:
   * If the shiftKey is **true**, get the token from the button's data-token attribute and transform it into its capitalized version (e.g., {he} becomes {He}).
   * If the shiftKey is **false**, use the token from data-token as is.
4. **Insert Correct Token**: Insert the correctly capitalized token into the modalTraitDescription textarea at the current cursor position.
5. **Remove Toggle Listener**: Delete the event listener associated with the now-removed trait-reference-toggle.